

**PROVA SCRITTA DEL CORSO DI
CALCOLATORI ELETTRONICI
NUOVO ORDINAMENTO DIDATTICO**
31 gennaio 2017

NOME:

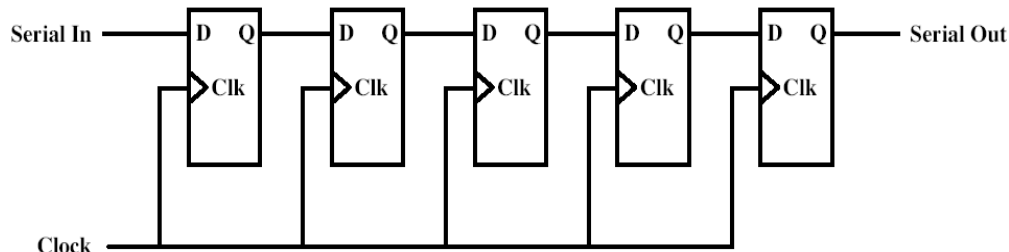
COGNOME:

MATRICOLA:

ESERCIZIO 1 (6 punti)

(3 punti) La seguente rete logica è combinatoria o sequenziale?

(3 punti) Qual è la funzione realizzata dalla rete?



ESERCIZIO 2 (8 punti)

Si consideri un calcolatore che dispone di una memoria principale di 256 Mbyte e di una memoria cache di 512 Kbyte. E' possibile accedere al singolo byte e la memoria è suddivisa in blocchi da 16 byte.

- 1) (4 punti) Spiegare, precisando il significato e la funzione dei diversi campi, come vengono interpretati gli indirizzi logici per recuperare l'informazione contenuta nella cache nel caso venga usata la modalità di indirizzamento
 - (a) Diretto
 - (b) "associativo su insiemi", e ciascun insieme contenga due blocchi
- 2) (4 punti) Si considerino le due parole di indirizzo rispettivamente 21A0x43 e 10y0F4D. Calcolare i valori delle cifre x e y tali per cui le due parole si trovino
 - (a) Nella stessa "linea" di cache nel caso di indirizzamento diretto
 - (b) Nello stesso insieme nel caso di indirizzamento associativo su insiemi a due vie

ESERCIZIO 3 (5 punti)

I trasferimenti di parole a/dalla memoria di un calcolatore sono codificati utilizzando il codice di Hamming. Si consideri la stringa di 12 bit 001001101110 (il bit meno significativo è a sinistra), risultata della codifica di una parola di N bit secondo il codice di Hamming.

- 1) (2 punti) calcolare N, supponendo di aver fatto uso del numero minimo di bit di controllo necessario per una stringa di 12 bit;
- 2) (3 punti) verificare la presenza di errori nella stringa data.

ESERCIZIO 4 (8 punti)

Indicare in modo chiaro e sintetico qual è la funzione svolta da ognuno dei tre frammenti MIPS seguenti. Si accetta come risposta anche la traduzione in linguaggio C dei frammenti. In ogni caso sia chiaramente indicata la funzione dei registri e delle aree di memoria.

Frammento 1 (3 punti).	Frammento 2 (2 punti).	Frammento 3 (3 punti).
<pre>mul \$15, \$16, 4 lw \$14, 1024(\$15) add \$14, \$14, \$13 subi \$15, \$15, 4 sw \$14, 1024(\$15)</pre>	<pre>move \$16, \$0 lui \$16, 1_10 addi \$16, \$16, 2305_10</pre>	<pre>Loop: add \$9, \$9, \$20 lw \$8, save(\$9) bne \$8, \$21, Loop</pre>

ESERCIZIO 5 (6 punti)

- (1) (3 punti) Indicare e disegnare uno schema di arbitraggio con tre linee disponibili per cinque periferiche indicando la funzionalità di ciascuna linea. Disegnare e spiegare in dettaglio, passaggio per passaggio, il protocollo di comunicazione che utilizzi tali linee.
- (2) (3 punti) Ipotizzare che i trasferimenti da calcolatore a periferica siano gestiti mediante modulo DMA. La linea dati è di 64 bit e la velocità del processore è pari a 1.4 GHz. Qual è la massima velocità di trasferimento ammissibile, espressa in MB/s?

ESERCIZIO 1

La rete è senza dubbio sequenziale, data la presenza di elementi di ritardo quali appunto i flip flop D, la cui tabella di eccitazione è:

$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

L'effetto che si ottiene dalla connessione seriale di tali flip flop, come in figura, è quello di far propagare, ad ogni colpo di clock, il bit di ingresso immesso nella linea "Serial input" verso i flip flop via via più "lontani" fino al raggiungimento della linea "Serial output", implementando un registro a scorrimento verso destra, equivalente ad una moltiplicazione per 2 del valore binario in esso contenuto.

ESERCIZIO 2

1) Per indirizzare 256 Mbyte occorre un indirizzo di almeno 28 bit. Per indirizzare il singolo byte all'interno di un blocco occorrono 4 bit ($16 = 2^4$), che coincidono con i 4 bit meno significativi dell'indirizzo di memoria primaria. I restanti 24 bit costituiscono l'indirizzo del "block frame". Per indirizzare la cache, il "block frame" viene interpretato diversamente a seconda che l'indirizzamento sia di tipo "diretto" o "associativo su insiemi".

(a) Indirizzamento diretto. In questo caso devo poter indirizzare ciascuno dei 32K blocchi contenuti nella cache ($512\text{Kbyte}/(16\text{byte}/\text{blocco})$). Occorrono 15 bit che coincidono con i 15 bit meno significativi del "block frame". Pertanto i 28 bit di indirizzo della memoria primaria vengono interpretati come:

tag	cache index	Offset
9 bit	15 bit	4 bit

(b) Indirizzamento "associativo su insiemi". In questo caso devo poter indirizzare ciascuno dei 16 insiemi in cui sono suddivisi i blocchi contenuti nella cache ($\frac{512\text{Kbyte}}{2\text{blocchi insieme} \cdot 16\text{byte blocco}} = 16\text{K}$ insiemi). Occorrono 14 bit che coincidono con i 14 bit meno significativi del "block frame"

Pertanto i 28 bit di indirizzo della memoria primaria vengono interpretati come:

tag	cache index	offset
10 bit	14 bit	4 bit

2. Perché le due parole si trovino nella stessa "linea" di cache o nello stesso insieme è necessario eguagliare l'index delle due parole. Faremo riferimento alla suddivisione in campi dell'indirizzo mostrato nella soluzione della prima parte dell'esercizio.

Nel caso di indirizzamento diretto avremo:

tag	cache index	offset
9 bit	15 bit	4 bit
001000011	0100000xxxx0100	0011
00010000y	yyy000011110100	1101

Cioè, in esadecimale, $x = F$, mentre y può essere pari a 2 od A perché la cifra più significativa è nel tag.

Analogamente, per il caso di indirizzamento associativo su insiemi a due vie, si ha:

tag	cache index	offset
10 bit	14 bit	4 bit
0010000110	100000xxxx0100	0011
00010000yy	yy000011110100	1101

Cioè, in esadecimale, $x = F$, mentre y può essere pari a 2, 6, A, E perché le due cifre più significative sono nel campo tag.

ESERCIZIO 3

1) Deve essere rispettata la condizione:

$$2^K \geq N + K + 1 \quad (1),$$

dove K è il numero di bit di controllo inseriti. Essendo $N + K = 12$, si evince dalla (1) che il numero minimo di bit di controllo richiesto è 4. Da cui $N = 8$.

2) Nella codifica di Hamming, la sequenza in ingresso presenta la seguente struttura:

1	2	3	4	5	6	7	8	9	10	11	12
c ₀	c ₁	b ₀	c ₂	b ₁	b ₂	b ₃	c ₃	b ₄	b ₅	b ₆	b ₇
0	0	1	0	0	1	1	0	1	1	1	0

Dove c₀...c₃ sono i quattro bit costituenti il vettore di controllo, e b₀...b₇ gli otto bit trasmessi. La sequenza ricevuta è 10111110.

Per verificare la presenza di eventuali errori, è sufficiente ricalcolare le parità tra i bit di informazione utilizzate per calcolare i bit di controllo; il successivo confronto attraverso distanza di Hamming con i bit di controllo pervenuti, permetterà di rilevare e localizzare l'eventuale bit errato.

C₀ = EXOR (b₀, b₁, b₃, b₄, b₆) = EXOR(1,0,1,1,1) = 0 == bit controllo pervenuto → 0

C₁ = EXOR (b₀, b₂, b₃, b₅, b₆) = EXOR(1,1,1,1,1) = 1 <> bit controllo pervenuto → 1

C₂ = EXOR (b₁, b₂, b₃, b₇) = EXOR(0,1,1,0) = 0 == bit controllo pervenuto → 0

C₃ = EXOR (b₄, b₅, b₆, b₇) = EXOR(1,1,1,0) = 1 <> bit controllo pervenuto → 1

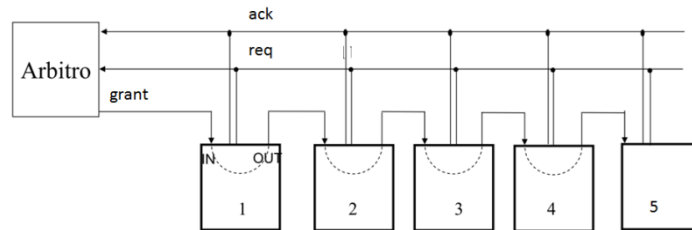
L'errore corrisponde alla configurazione 1010 → 8 + 2 = 10 → bit b₅ che va commutato a 0.

ESERCIZIO 4

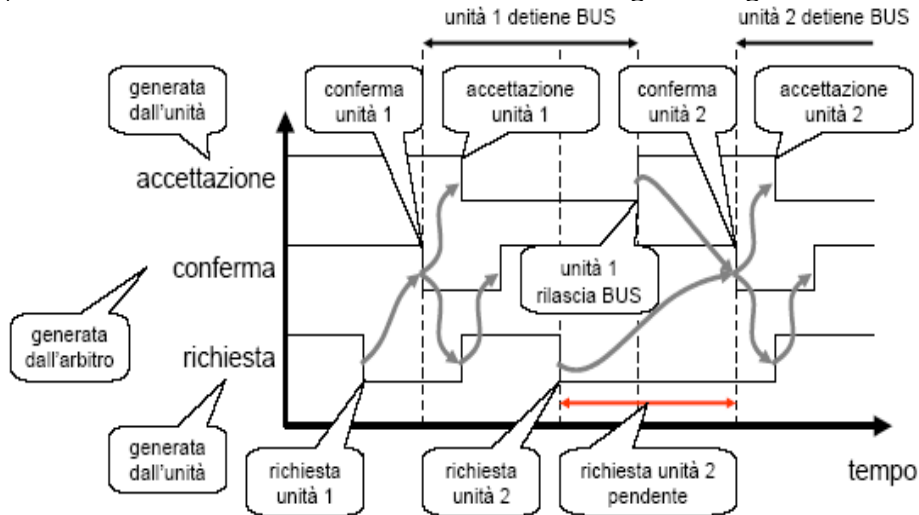
Frammento 1. mul <i>\$15</i> , <i>\$16</i> , 4 lw <i>\$14</i> , 1024(<i>\$15</i>) add <i>\$14</i> , <i>\$14</i> , <i>\$13</i> subi <i>\$15</i> , <i>\$15</i> , 4 sw <i>\$14</i> , 1024(<i>\$15</i>)	Frammento 2. move <i>\$16</i> , <i>\$0</i> lui <i>\$16</i> , 1 ₁₀ addi <i>\$16</i> , <i>\$16</i> , 2305 ₁₀	Frammento 3. Loop: add <i>\$9</i> , <i>\$9</i> , <i>\$20</i> lw <i>\$8</i> , save(<i>\$9</i>) bne <i>\$8</i> , <i>\$21</i> , Loop
A[i-1]=A[i]+h; Con: i → <i>\$16</i> h → <i>\$13</i> A == 1024	Inserimento nella parte più significativa di <i>\$16</i> della costante 1 ed in quella meno significativa di 2305, ottenendo così il valore a 32 bit pari a 2 ¹⁶ +2305.	Ciclo o loop: do i=i+j; while (save[i]!=h); con: i → <i>\$9</i> j → <i>\$20</i> h → <i>\$21</i>

ESERCIZIO 5

- 1) Un possibile schema di arbitraggio è quello centralizzato con tre linee GRANT, ACK, REQ, la prima delle quali attivata dall'arbitro e le altre due dalle periferiche. Lo schema corrispondente è il seguente:



Il relativo protocollo di comunicazione è descritto dalla seguente figura:



In questa figura i segnali sono "attivi bassi". Quando l'Unità 1, ad esempio, compie la richiesta, essa deve attendere dall'arbitro il "grant" ovvero la conferma che, non appena avvenuta, consente alla periferica di attivare la linea "ack" (che significa anche bus "impegnato"). Successivamente l'arbitro disattiva la linea "grant" e la periferica disattiva la linea "req". L'attivazione della linea "ack" impedisce che la richiesta di una seconda periferica, ad esempio la 2, possa essere presa in considerazione dal bus fino a che quella correntemente in possesso non lo rilasci disattivando la linea "ack". A quel punto, l'arbitro riattiverà la linea "grant" consentendo alla prima periferica del festone che ha richiesto il bus di utilizzarlo. Si ricordi infatti che il segnale "grant" si propaga sulle periferiche nell'ordine cablato dalla "daisy chain".

- 2) Nel caso di massima velocità, il DMA deve lavorare in modalità "block transfer" che garantisce un trasferimento dati ad ogni ciclo di clock.
Poiché la linea dati consente il trasferimento di 8 byte, il calcolo è dato da : $1.4 \cdot 10^9$ cicli al secondo $\cdot 8$ byte/ciclo = $11,2 \cdot 10^9 / 2^{20} = 10681$ MB/sec